ML2020SPRING HW14 Report

學號：R08946015 系級：資料科學碩一 姓名：陳鈞廷

1. 請以中文說明一下 lifelong learning 的中心概念是什麼?

以往我們在使用每個 machine learning 來解決不同的 task 時，都是針對不同的 task 來設計模型架構，並且**從頭**訓練模型，然而這樣就代表每個 task 都需要自己擁有個別的模型架參數甚至是模型架構，而且要每個 task 都從頭訓練模型會非常耗費時間。因此我們希望可以建立同一個 machine learning 演算法來解決不同的 task，因為我們認為模型在過去學習其他 task 的時候已經累積了不少經驗與知識，而這些經驗與知識應該可以幫助解決新的 task 的學習。簡而言之就是希望模型可以 (1) 持續不斷學習，(2) 累積過去學習過的 task 中的經驗與知識，(3) 以此幫助解決新的 task 學習。

2. 列出 EWC, MAS 的作法是什麼?根據你的理解， 說明一下大概的流程該怎麼做。

在 lifelong learning 中需要解決 model 在學習新的 task 時會遺忘過去的知識的問題，假設 model 先學習了 task 1 後再去學習 task 2，因為 task 1 跟 task 2 的 loss 的分佈不太相同，所以可能會發生 model 在學習 task 2 時候 task 1 的 loss 反而往上漲的情況，因此我們希望 model 在學習完 task 2 後的參數不要和剛學完 task 1 的參數相差太多，確保 model 在在學習完 task 2 後還能在 task 1 有不錯的 performance。

而 EWC 和 MAS 都是 regularization based 的方法，在 loss function 中加上對參數的限制，使某些在過去 task 中學習到的重要參數不會有太多的更新，因此一定程度的確保了過去 task 的 performance。如下方列式，$L(\theta)$ 是 task 2 原本的 loss function，而 θ^b 是學習完 task 1 的 parameter，在 EWC 和 MAS 中就是在 loss function 中加上 regularization term，其中我們需要決定 λ 的大小，太大的 λ 反而會讓 model 沒辦法學習到 task 2；另外我們需要在學習的過程中來計算 b_i 來決定哪個參數是重要的，重要的參數需要比較大的 b_i 來限制著參數的更新。

$$L'(\theta) = L(\theta) + \lambda \sum_i b_i \left(\theta_i - \theta_i^b\right)^2$$

一開始我們希望從 loss function 的二階微分來得知哪些 parameter 對於 loss 是重要的，但實作上計算二階微分比較麻煩，因此在 **EWC** 的實作上會使用 **Fisher information matrix** 來替代，也就是說上面等式中的 b_i 就是計算出來的 fisher information matrix 的對角線值。

MAS 跟 EWC 的差異主要在 b_i 的計算上，**MAS** 在實作上首先計算上個 task 儲存的 model 的 output vector 的平方和，再將這個平方和對參數計算 gradient 後再取絕對值後就完成 MAS 的 b_i 計算了。

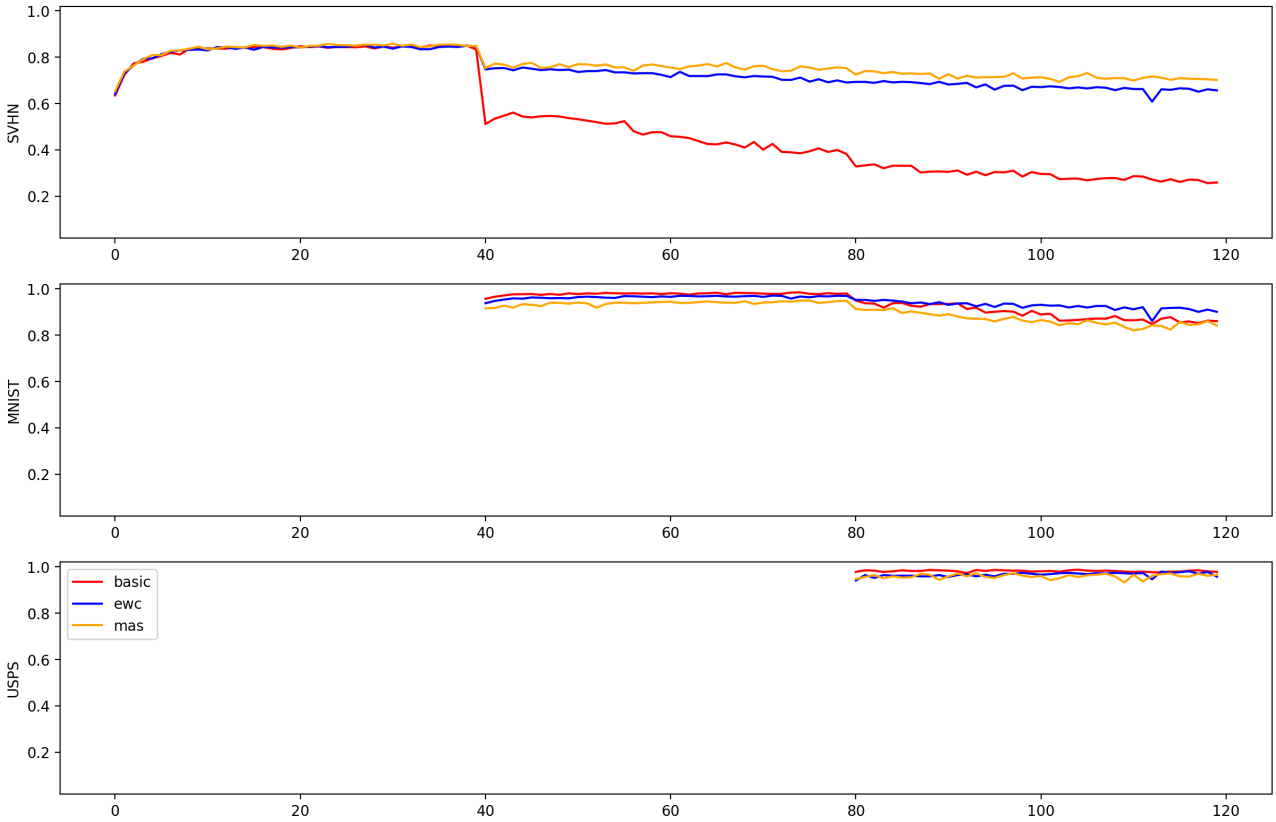
3. EWC 和 MAS 方法上所需要的資料最大的差異是什麼

EWC 在計算 fisher information matrix 的時候會先計算 posterior probability，因此需要前一個 task 的 training data 以及 label ；而 MAS 因為是計算前一個 model 的 output vector，所以只需要前一個 task 的 training data，並不需要相對應的 ground truth label。

4. 秀出 part1 及 part2 最後結果的圖，並分析一下結果，以及你跑的實驗中有什麼發現

Part1

經過數次調整 regularization term 的 λ，選定 **EWC** 的 $\lambda = 400$，**MAS** 的 $\lambda = 0.1$，(Basic 並沒有 λ)。實驗結果如下圖所示，可以發現紅色線的 basic 在開始學習 MNIST 後，SVHN 的 accuracy 急劇下降，但 EWC 和 MAS 的下降幅度相對不明顯；比較有趣的是 MAS 的 MNIST 的 accuracy 是最差的，這有可能是因為 regularization 的約束效果太強，使得 MAS 在 MNIST 的學習比其他兩個來的差，而這個差異在開始學習 USPS 後變得更明顯。



Part2

這裡我選擇實作 SCP，設定 SCP 的 $\lambda = 10$，其他三個演算法設定與 part1 相同，實驗結果如下圖所示。 從途中可以發現 SCP 的效果與 EWC 差不多，而且在 MNIST 上也學得比 MAS 來得好，因此推測在這樣的 λ 設定下，SCP 的 regularization 約束效果比 MAS 來的弱。總的來說，EWC、MAS 和 SCP 都是有效的 lifelong learning 演算法，但跟據 λ 的設定，可能會出現有 0.05 左右的 accuracy 差異。